UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/840,164	05/06/2004	Nicolai Kosche	188378/US/2	7512

66083 7590 10/16/2008

SUN MICROSYSTEMS, INC. c/o DORSEY & WHITNEY, LLP
370 SEVENTEENTH ST.
SUITE 4700
DENVER, CO 80202

EXAMINER	
TECKLU, ISAAC TUKU	
ART UNIT	**PAPER NUMBER**
2192	
MAIL DATE	**DELIVERY MODE**
10/16/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/840,164	KOSCHE ET AL.	
	Examiner	Art Unit	
	ISAAC T. TECKLU	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-58* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-58* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-58 have been examined.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1-15 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 1 is non-statutory as being "A computer-implemented software tool" without being supported by hardware such as tangible computer storage or execution engine, which would enable one skill in the art to construe that the software tool is built from tangible product to carry out any functionality being conveyed from the claim. Thus, the computer-implemented software tool is computer listings *per se*, i.e., the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer-implemented software tool do not define any structural and functional interrelationships between the computer-implemented software tool and

other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer- readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. See Lowry, 32 F.3d at 1583-84, 32 USPQ2d at 1035. Accordingly, it is important to distinguish claims that define descriptive material per se from claims that define statutory inventions.

Claims 2-15 are rejected for failing to cure the deficiencies of the above rejected non-statutory claim 1 above. See MPEP 2106.01(I).

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-58 are rejected under 35 U.S.C. 102(e) as being anticipated by Yates, Jr. et al. (US 7,111,290 B1), hereinafter Yates.

As per claim 1(Currently Amended), Yates discloses a computer-implemented software tool that determines at least one data address from one or more instruction instances (col. 2:15-25 "… figures out the address …" and col. 7:15-40 "… execution speed … instance of instruction have an event code that leaves intact an event code previously determined…" and see at least col.6:60-67 and col.7:1-10), and that identifies one or more memory reference objects, associated with the data address, as hindering execution of code that includes the instruction instances, wherein the instructions instances correspond to the code execution hindrance (col. 6:51-59 "… profile information is recorded that records physical memory reference …" and col. 7:15-40 "… execution speed … instance of instruction have an event code that leaves intact an event code previously determined …" and see also col.17:30-50).

As per claim 2, Yates discloses the software tool of claim 1 wherein the memory reference objects include one or more of physical memory reference objects and logical memory reference objects (col. 7:14-25 "… physical memory reference … reference may record the event of a sequential execution flow…" and col. 17:15-20 "… memory references referring to logical address …).

As per claim 3, Yates discloses the software tool of claim 2 wherein the physical memory

reference objects include one or more of cache (e.g. Fig. 1a, 112 and related text), cache lines (col.88: 15-20 ".. cache lines…"), cache levels (e.g. Fig. 1c, DATA CACHE and related text), cache sub-blocks (e.g. Fig. 1c, 112 and related text), memory controllers (e.g. Fig. 1a, 124 and memory modification monitor and related text), addressable memory (e.g. Fig. 1c, 146 and related text), and memory-management page translation units (e.g. Fig. 1a, 170 and related text).

As per claim 4, Yates discloses the software tool of claim 3, wherein the addressable memory includes one or more of virtually addressable memory and physically addressable memory (col. 29: 1-10 "… virtual address …").

As per claim 5, Yates discloses the software tool of claim 2 wherein the logical memory reference objects include one or more of source-level data objects, memory segments (e.g. Fig. 1D and related text), heap variables (col. 30:40-50 "heap"), variable instances (col. 30:40-50 "state variable"), and stack variables (col. 30:40-50 "stack ").

As per claim 6, Yates discloses the software tool of claim 5 wherein the source-level data objects include one or more of functions (e.g. Fig. 1e and related text), statically linked objects (e.g. Fig. 1e, 56 and related text), data structures (e.g. Fig.1e, 48 and related text), data types (e.g. Fig. 1e, 40 and related text), data type definitions (e.g. Fig. 1e, 32 and related text), operands (e.g. Fig. 3g, 317 and related text), and expressions (e.g. Fig. 3g, 317 and related text).

As per claim 7, Yates discloses the software tool of claim 6 wherein the statically linked

objects include one or more of global variables and static variables (e.g. TABLE 1 and related text).

As per 8, Yates discloses the software tool of claim 1 wherein the software tool includes one or more of a compiler, an interpreter (col. 19:20-40 "… interpreter…"), an optimization tool (col.19:20-30 "… emulator…"), and a virtual machine (e.g. Fig. 1a, 118 and related text).

As per claim 9, Yates discloses the software tool of claim 1 wherein the code includes one or more of machine code (e.g. Fig. 1a, 118 and related text), byte code (col.19:20-30 "… emulator…"), and interpreted code (col. 19:20-40 "… interpreter…").

As per claim 10, Yates discloses the software tool of claim 1 that also aggregates addresses based on the memory reference objects (e.g. Fig. 1c and related text).

As per claim 11, Yates discloses the software tool of claim 10 wherein the software tool utilizes at least a portion of the data addresses to aggregate the addresses (e.g. Fig. 1c and related text).

As per claim 12, Yates discloses the software tool of claim 10 that also provides the aggregated addresses and an indication of the code execution hindrance corresponding to the aggregated addresses for one or more of storage and display (e.g. Fig. 1c and related text)..

As per claim 13, Yates discloses the software tool of claim 1 wherein the data address includes a virtual address or a physical address (e.g. Fig. 1d and related text)..

As per claim 14, Yates discloses the software tool of claim 1 wherein the code execution hindrance corresponds to one or more sampled runtime events (e.g. Fig. 1d and related text)..

As per claim 15, Yates discloses the software tool of claim 14 wherein the sampled runtime events include one or more of cache misses, cache references, data translation buffer misses, data translation buffer references, and counter condition events (e.g. Fig. 1b and related text).

As per claim 16, Yates discloses a method for profiling code, the method comprising:

identifying an instruction instance that corresponds to a runtime event col. 6:51-59 "… profile information is recorded that records physical memory reference …" and col. 7:15-40 "… execution speed … instance of instruction have an event code that leaves intact an event code previously determined …");

determining a data address from the instruction instance (col. 2:15-25 "… figures out the address …"); and

determining a memory reference object from the determined address (col. 7:14-25 "… physical memory reference … reference may record the event of a sequential execution flow…" and col. 17:15-20 "… memory references referring to logical address …).

As per claim 17, Yates discloses the method of claim 16 wherein the runtime event is a sampled runtime event (e.g. Fig. 6c, 650 and related text).

As per claim 18, Yates discloses the method of claim 16 wherein identifying the instruction
instance comprises backtracking from a second instruction instance to the instruction instance (e.g. Fig. 650 and related text).

As per claim 19, Yates discloses the method of claim 16 wherein determining the address from the instruction instance comprises decoding the instruction instance (e.g. Fig. 5b, 556 and related text).

20. The method of claim 19 further comprising:

decoding the instruction instance if a register that hosts the instruction instance is determined as valid (e.g. Fig. 5b, 556 and related text).

As per claim 21, Yates discloses the method of claim 20 wherein determining if the register is valid comprises:

applying reverse register transformation with respect to the runtime event (e.g. Fig. 3d and related text); and

determining whether the register is valid based on the applied reverse register transformation (e.g. Fig. 1d and related text).

As per claim 22, Yates discloses the method of claim 16 wherein the memory reference object includes a physical memory reference object or a logical memory reference object (col. 7:14-25 "… physical memory reference … reference may record the event of a sequential execution flow…" and col. 17:15-20 "… memory references referring to logical address …").

As per claim 23, this is the method version of the claimed software tool discussed above (Claim 3), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 24, this is the method version of the claimed software tool discussed above (Claim 4), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 25, this is the method version of the claimed software tool discussed above (Claim 6), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 26, this is the method version of the claimed software tool discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 27, this is the method version of the claimed software tool discussed above (Claim 8), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 28, this is the method version of the claimed software tool discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 29, this is the method version of the claimed software tool discussed above (Claim 10), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 30, this is the method version of the claimed software tool discussed above (Claim 11), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 31, this is the method version of the claimed software tool discussed above (Claim 12), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 32, this is the computer program product version of the claimed method discussed above (Claim 16), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 33, Yates discloses a method of profiling code, the method comprising:

associating data addresses with memory reference objects, wherein the data addresses have been determined from instruction instances corresponding to code execution hindrance (col. 2:15-25 "… figures out the address …"); and

aggregating the data addresses based on their associated memory reference objects (e.g. Fig. 1c and related text).

As per claim 34, this is the method version of the claimed software tool discussed above (Claim 8), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 35, this is the method version of the claimed software tool discussed above (Claim 14), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 36, this is another method version of the claimed method discussed above (Claim 17), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 37, this is another method version of the claimed software tool discussed above (Claim 15), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 38, this is another method version of the claimed software tool discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 39, this is another method version of the claimed software tool discussed above (Claim 11), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 40, Yates discloses this is the computer program product version of the claimed method discussed above (Claim 33), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 41, Yates discloses a method of profiling code comprising:

identifying an instruction instance corresponding to a runtime event (col. 2:15-25 "… figures out the address …" and col. 7:15-40 "… execution speed … instance of instruction have an event code that leaves intact an event code previously determined…");

determining whether the instruction instance is valid (col. 6:51-59 "… profile information is recorded that records physical memory reference …" and col. 7:15-40 "… execution speed … instance of instruction have an event code that leaves intact an event code previously determined …");

decoding the instruction instance to extract at least a portion of a data address(e.g. Fig. 5b, 556 and related text)

if the instruction instance is valid (e.g. Fig. 5b, 556 and related text);

determining a memory reference object with the extracted portion of the address (col. 7:14-25 "… physical memory reference … reference may record the event of a sequential execution flow…" and col. 17:15-20 "… memory references referring to logical address …); and

aggregating the data address with other addresses based at least in part on the memory reference object (e.g. Fig. 1c and related text).

As per claim 42, this is another method version of the claimed software tool discussed above (Claim 10), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 43, this is another method version of the claimed software tool discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 44, this is another method version of the claimed software tool discussed above (Claim 14), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 45, this is another method version of the claimed method discussed above (Claim 21), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 46, this is the computer program product version of the claimed method discussed above (Claim 41), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 47, this is the computer program product version of the claimed method discussed above (Claim 41), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 48, this is the computer program product version of the claimed method discussed above (Claim 42), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 49, this is the computer program product version of the claimed method discussed above (Claim 45), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 50, this is the computer program product version of the claimed software tool discussed above (Claim 13), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 51, this is the computer program product version of the claimed software tool discussed above (Claim 15), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 52, this is the computer program product version of the claimed software tool discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 53, this is the computer program product version of the claimed software tool discussed above (Claim 6), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 54, this is computer program product version of the claimed software tool discussed above (Claim 7), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 55, this is the apparatus version of the claimed method discussed above (Claim 41), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 56, this is the apparatus version of the claimed method discussed above (Claim 42), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 57, this is apparatus version of the claimed software tool discussed above (Claim 5), wherein all claim limitations have been addressed and/or covered in cited areas as set forth above. Thus, accordingly, these claims are also anticipated by Yates.

As per claim 58, Yates discloses the apparatus of claim 56 wherein the processor includes event condition counters (col. 2:15-25 "… counters will indicate …").

Response to Arguments

6. Applicant's arguments filed 03/25/08 have been fully considered but they are not persuasive.

A. The Applicant argued: "Yates fails to anticipate the independent claims because Yates does not disclose determining at least one data address from one or more instruction instances (page 11).

The Examiner cited particular columns and line numbers in the references as applied to the claims **for the convenience of the applicant**. Although the specified citations are representative of the teachings in the art and are applied to the specific limitations within the individual claims, other passages and figures may apply as well. It is respectfully requested from the applicant, in preparing any response, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the examiner.

The Examiner disagrees. For instance, Yates discloses determining at least one data address from one or more instruction instances below:

"In general, in a thirteenth aspect, a program is executed on a computer, the program referring to memory by virtual address. Concurrently with the execution of the program, profile information is recorded describing memory references made by the program, the profile information recording physical addresses of the profiled memory references" (col.6:60-67 and col.7:1-10, emphasis added).

"The recorded physical memory references may include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references may record the event of a sequential execution flow across a page boundary in the address space. The recorded execution flow across a page boundary may occur within a single instruction. The recorded execution flow across a page boundary may occur between two instructions that are sequentially adjacent in the logical address space. At least one of the recorded instruction references may be a divergence of control flow consequent to an external interrupt. At least one of the recorded instruction references may indicate the address of the last byte of an instruction executed by the computer during the profiled execution interval." (col.7:15-40, emphasis added).

"The invention features a method and computer for performance of the method. Memory references generated as part of executing a stream of instructions on a computer are evaluated to determined whether an individual memory reference of an instruction references a device having a valid memory address." (col.17:50-60, emphasis added).

B. The Applicant argued: "Yates fails to anticipate the independent claims because Yates does not disclose identifying one or more memory reference objects, associated with the data address (page 11-12).

The Examiner disagrees. For instance, Yates teaches identifying one or more memory reference objects, associated with the data address below:

"In general, in a thirteenth aspect, a program is executed on a computer, the program referring to memory by virtual address. Concurrently with the execution of the program, profile information is recorded describing memory references made by the program, the profile information recording physical addresses of the profiled memory references." (col.6:60-67 and col.7:1-10, emphasis added).

"Embodiments of the invention may include one or more of the following features. The recorded physical memory references may include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references may record the event of a sequential execution flow across a page boundary in the address space. The recorded execution flow across a page boundary may occur within a single instruction. The recorded execution flow across a page boundary may occur between two instructions that are sequentially adjacent in the logical address space." (col.7:15-40, emphasis added).

"The form of the recording may indicates an address of an instruction that issued the memory reference. The memory reference may be a load. The profile monitoring circuitry may be interwoven with the computer CPU. A TLB (translation look aside buffer) may be designed to hold a determination of whether memory mapped by entries of the TLB is well-behaved or

non-well-behaved memory. The profile monitoring circuitry may generate the record into a general purpose register of the computer. The profile monitoring circuitry may be designed to induce a pipeline flush of the computer CPU." (col.17:30-50, emphasis added).

"In general, in a forty-seventh aspect, the invention features a method and computer for performance of the method. Memory read references are generated in a CPU of a computer, the memory references referring to logical addresses. Circuitry and/or software evaluates whether main memory pages of the references are in a protected state. Pages that are unprotected are put into a protected state." (col.17:5-15, emphasis added).

"In general, in a forty-ninth aspect, the invention features a method and computer for performance of the method. Memory references generated as part of executing a stream of instructions on a computer are evaluated to determined whether an individual memory reference of an instruction references a device having a valid memory address but that cannot be guaranteed to be well-behaved." (col.17:50-60, emphasis added).

Thus, it is respectfully submitted that the claims will stand rejected as set forth in the Office Action.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571)272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192